REMARKS

This Amendment is filed in response to the Office Action mailed January 8, 2010. All objections and rejections are respectfully traversed.

Claims 1-11, 17-34, and 36-41 are currently pending.

No new claims have been added.

No claims have been amended.

Interview Summary

Applicant would like to thank Examiner Morrison for conducting the Applicant Initiated Interview on March 5, 2010 and for helping to advance this Application closer to allowance. Generally, as will be elaborated upon in greater detail below, the issue discussed involved Applicant's use of **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. Specifically, Applicant discussed that while Lou may or may not teach deleting something from a hash table, Lou does not teach deleting something from the hash table in response to a match of the hashed (description tag) value (e.g., "mousebuttontest"). Instead, Lou teaches hashing a value (i.e., a hashed description tag value) and then comparing an associated non-hashed results value (i.e., pass, fail, 0, 1, etc.). If the non-hashed results value is matched, then the non-hashed results value is deleted from the hash table. Thus, Applicant discussed that Lou fails to teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value*** of the selected entry of the first data set from the hash table **in response to determining** that ***the hashed value*** of the selected entry of the first data set **is in the hash table**.

While Examiner initially agreed with Applicant, Examiner noted that a closer look at the prior art would be required to verify Applicant's contentions and that another search would be required. Examiner is encouraged to contact the undersigned attorney with any questions. Examiner also noted that if a new search resulted in a new discovery of relevant art, Examiner would contact the undersigned attorney to discuss the art before issuing the next Office Action.

Rejections Under 35 U.S.C. §103

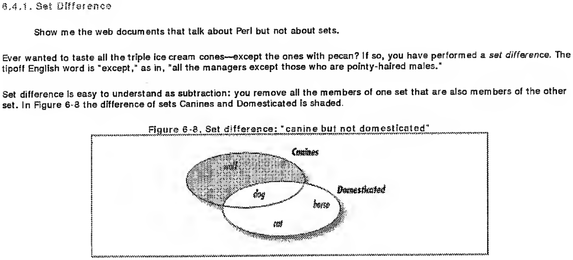
At paragraph 4 of the Office Action, claims 22-28 were rejected under 35 U.S.C. §103(a) as being obvious over Orwant et al., "*Mastering Algorithms with Perl*" (hereinafter "Orwant"), in view of Musser, "*Rationale for Adding Hash Table to the C++ Standard Template Library*" (hereinafter "Musser"), and in further view of Lou, U.S. Patent No. 7,096,421 issued on August 22, 2006 (hereinafter "Lou").

Applicant's claimed novel and non-obvious invention, as set forth in representative claim 22, comprises in part:

> 22. A method for comparing a first data set with a second data set, comprising:
>
> (a) selecting an entry from the first data set, wherein the first data set is stored on a source storage system;
>
> (b) determining if a hashed value of the selected entry of the first data set is in a hash table, wherein the hash table comprises one or more hashed values of the first data set;
>
> (c) adding, in response to determining that the hashed value of the selected entry of first data set is not in the hash table, the hashed value of the selected entry of the first data set to the hash table;
>
> (d) **removing from the hash table, in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table, *the hashed value* of the selected entry of the first data set**;
>
> (e) selecting an entry from the second data set, wherein the second data set is stored on a destination storage system;
>
> (f) determining if a hashed value of the selected entry of the second data set is in the hash table, wherein the hash table further comprises one or more hashed entries of the second data set;
>
> (g) adding, in response to determining that the hashed value of the selected entry of the second data set is not in the hash table, the hashed value of the selected entry of the second data set to the hash table;
>
> (h) removing from the hash table, in response to determining that the hashed value of the selected entry of the second data set is in the hash table, the hashed value of the selected entry of the second data set;

> (i) continuing (a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed; and
>
> (j) reporting a difference between the first data set and the second data set in response to at least one hashed value remaining in the hash table.

Orwant teaches, in relevant part as cited by Examiner, "set difference". In particular, Orwant explicitly describes set difference at section 6.4.1 as follows:



6.4.1. Set Difference

Show me the web documents that talk about Perl but not about sets.

Ever wanted to taste all the triple ice cream cones—except the ones with pecan? If so, you have performed a *set difference*. The tipoff English word is "except," as in, "all the managers except those who are pointy-haired males."

Set difference is easy to understand as subtraction: you remove all the members of one set that are also members of the other set. In Figure 6-8 the difference of sets Canines and Domesticated is shaded.

Figure 6-8. Set difference: "canine but not domesticated"

Orwant goes on further to explicitly define set difference as "noncommutative or asymmetric: that is, if you exchange the order of the sets, the result will change."

Musser teaches, in relevant part as cited by Examiner, a hash table (page 8, paragraph 9).

Lou teaches, in relevant part, comparing test reports of XML documents. Specifically, when a plurality of (e.g., Java script) tests are conducted, each individual test is given a description tag (e.g., "mousebuttontest"). <u>The description tag is hashed</u> to identify a location in a hash table. A <u>result tag</u> associated with the description tag includes a result value that identifies the results of the test (e.g., pass, fail, 1, 0, etc) in the hash table identified by the hashed description tag (col. 9, lines 42-55).

When two XML documents are compared (e.g., XML version 1.0 and XML version 1.1), their respective hashed description tags are hashed and compared. If their results are identical, then the same test (i.e., "mousebuttontest") was performed by each XML version. Next, each respective results value is compared. If the result values do not match, then one XML version failed the test and the other XML version passed the test (col. 10, lines 11-20). However, if the result values do match, then both XML versions either failed or passed the test. As a result, the data (i.e., **the corresponding results value**) is deleted from the identified location in the hash table (col. 10, lines 21-30).

Applicant respectfully urges that Orwant, taken singly or in any combination with Musser and/or Lou, does not disclose Applicant's claimed novel and non-obvious use of

removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table.

Applicant claims, in part, selecting an entry from a first data set and then determining if a hashed value of that selected entry is in a hash table, wherein the hash table comprises one or more hashed values from the first data set. In other words, broadly speaking, Applicant claims hashing entries of a first data set, storing one or more of the hashed values of the hashed entries in a hash table, and then determining if a particular hashed value of the first data set matches one of those stored hash values located in the hash table. **In response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**, Applicant claims **removing *the hashed value* of the selected entry of the first data set from the hash table**.

As an aside, Applicant respectfully argued in the FINAL Amendment filed on October 21, 2009 that Orwant's "set difference" and/or "members" were not the same as Applicant's claimed data sets and/or entries. Applicant maintains these contentions; however, in the interest of advancing prosecution and avoiding the delay in seeking appellate review from the Board of Patent Appeals and Interferences and/or the U.S. Court of Appeals for the Federal Circuit, Applicant respectfully presents an alternative argument. Applicant expressly reserves the right to present these contentions or variations thereof in any appellate procedures.

Applicant respectfully contends that Orwant does not teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. Specifically, as noted by Examiner on page 5 of the Office Action, Orwant does not explicitly indicate "hash" or "hash table". As such, because Orwant does not teach or suggest a "hash" or a "hash table", Orwant fails to teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table.**

Applicant respectfully contends that Musser does not teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. Specifically, as noted by Examiner on page 5 of the Office Action, neither Musser (nor Orwant) explicitly indicate "removing from the table, in response to determining that the value of the selected entry of the second data set is in the table, the hash value of the selected entry of the second data set". As such, because neither Musser (nor Orwant) teach or suggest "removing from the table, in response to determining that the value of the selected entry of the second data set is in the table, the hash value of the selected entry of the second data set", both Musser and Orwant fail to teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**.

Applicant respectfully contends that Lou does not teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. Specifically, as noted above, Lou teaches hashing a description tag (e.g., "mousebuttontest") to identify an entry location in a hash table, but Lou also explicitly states that it is the corresponding results value (tag) (e.g., pass, fail, 1, 0, etc) that is deleted from the identified location in the hash table (see Lou, at least in part, at col. 10, lines 26-29 below):

> In operation 814, the data (i.e., the corresponding **results value**) **is deleted** from the identified location in the hash table and the process continues in operation 818. (emphasis added)

Notably, the results value is not disclosed to be hashed. In other words, Lou teaches deleting a non-hashed results value, but Lou fails to teach or suggest deleting the actual hashed value of the entry in the hash table. In contrast, Applicant claims **removing *the hashed value* of the selected entry from the hash table** in response to determining that the hashed value of the selected entry is in the hash table. As such, because Lou fails to teach or suggest deleting the actual hashed value of the entry in the hash table, Lou fails to teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**.

However, even if it is assumed *arguendo* that Lou did show deleting the actual hashed value of the entry in the hash table, Lou still fails to teach or suggest Applicant's claimed novel and non-obvious removing the hashed value of the selected entry of the first data set from the hash table **in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. Specifically, as noted above, Lou explicitly teaches that a deletion from the hash table (regardless of what is deleted from the hash table), is deleted in response to a match between the result values (e.g., pass, fail, 1, 0, etc) of the identified entry locations (see Lou at col. 10, lines 21-30 below):

> Returning to operation 810, if the [corresponding results] value of the selected tag is equal to the [results] value [stored] in the identified location in the hash table, the common test case has the same value in both the first test report and the second test report. If the common test case has the **same [result] value[s]** in both the first test report and the second test report, the process continues in operation 814. In operation 814, the data (i.e., **the corresponding results value**) **is deleted** from the identified location in the hash table and the process continues in operation 818. (emphasis added)

As such, Lou explicitly teaches that a deletion from the hash table (regardless of what is being deleted from the hash table) is deleted *in response to* a match between the (non-hashed) result values and *not in response to* a match between the hashed description tag value. In other words, while Lou may or may not teach deleting something from a hash table, Lou does not teach deleting something from the hash table in response to a match of the hashed (description tag) value. Instead, Lou teaches hashing a value (i.e., the hashed description tag value) and then comparing an associated non-hashed results value. If the non-hashed results value is matched, then the non-hashed results value is deleted from the hash table. In contrast, Applicant claims **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**). Thus, Lou still fails to teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value*** of the selected entry of the first data set from the hash table **in response to determining** that ***the hashed value*** of the selected entry of the first data set **is in the hash table.**

Accordingly, Applicant respectfully urges that Orwant, taken singly or in any combination with Musser and/or Lou, is legally insufficient to render the presently claimed invention obvious under 35 U.S.C. §103. Orwant and/or Musser and/or Lou, taken singly or in any combination, does not disclose Applicant's claimed novel and non-obvious use of **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**.

At paragraph 5 of the Office Action, claims 1-11, 17-21, 29-31, and 39 were rejected under 35 U.S.C. §103(a) as being obvious over Orwant, in view of Musser, in further view of Rsync, "*Rsync Unix command manual*" version 2.4.1 (hereinafter "Rsync"), and in further view of Lou.

Applicant's claimed novel and non-obvious invention, as set forth in representative claim 1, comprises in part:

> 1. A method for comparing a first directory comprising unique elements with a second directory comprising unique elements, comprising:
>
> (a) for each entry in the first directory, placing a hash value of the entry in a hash table, wherein the first directory is stored on a source storage system;
>
> (b) selecting an entry of the second directory, wherein the second directory is located on a destination storage system;
>
> (c) looking up a match between a hash value of the selected entry and the hash value of the entry in the hash table;
>
> (d) **removing, in response to the match between *the hash value* of the selected entry and *the hash value* of the entry in the hash table, *the hash value* of the entry from the hash table**;
>
> (e) determining if an additional second directory entry exists;
>
> (f) looping to step (b) in response to identifying the additional second directory entry; and
>
> (g) reporting a difference between the first directory and the second directory in response to at least one hash value entry remaining in the hash table.

Rsync teaches, in relevant part as cited by Examiner, a directory (OPTIONS section). However, Rsync is silent to Applicant's claimed novel and non-obvious **removing, in response to the match between *the hash value* of the selected entry and *the hash value* of the entry in the hash table, *the hash value* of the entry from the hash table**.

Additionally, as noted above with regard to claim 22, neither Orwant nor Musser nor Lou teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. As such, because claim 1 comprises similar limitations of claim 22 not shown or suggested by any prior art reference (or combination thereof), Applicant respectfully urges that Rsync, taken singly or in any combination with Orwant and/or Musser and/or Lou, fails to render the presently claimed invention obvious under 35 U.S.C. §103(a). Specifically, Rsync and/or Orwant and/or Musser and/or Lou, taken singly or in any combination, do not disclose, teach or suggest Applicant's claimed novel and non-obvious use of **removing, in response to the match between *the hash value* of the selected entry and *the hash value* of the entry in the hash table, *the hash value* of the entry from the hash table.**

At paragraph 6 of the Office Action, claims 32-34, 36-38, and 40-41 were rejected under 35 U.S.C. §103(a) as being obvious over Rsync, in view of Musser, in view of Orwant, and in further view of Lou.

Applicant's claimed novel and non-obvious invention, as set forth in representative claim 37, comprises in part:

> 37. A computer readable medium containing executable program instructions executed by a processor, comprising:
>
> (a) program instructions that select an entry from a first data set, wherein the first data set is stored on a source storage system;
>
> (b) program instructions that determine if a hashed value of the selected entry of the first data set is in a hash table, wherein the hash table comprises one or more hashed values of the first data set;
>
> (c) program instructions that add, in response to determining that the hashed value of the selected entry of first data set is not in the hash table, the hashed value of the selected entry of the first data set to the hash table;
>
> (d) **program instructions that remove from the hash table, in response to determining that *the hashed value* of the selected entry of**

> **the first data set is in the hash table, *the hashed value* of the selected entry of the first data set;**
>
> (e) program instructions that select an entry from a second data set, wherein the second data set is stored on a destination storage system;
>
> (f) program instructions that determine if a hashed value of the selected entry of the second data set is in the hash table, wherein the hash table further comprises one or more hashed entries of the second data set;
>
> (g) program instructions that add, in response to determining that the hashed value of the selected entry of the second data set is not in the hash table, the hashed value of the selected entry of the second data set to the hash table;
>
> (h) program instructions that remove from the hash table, in response to determining that the hashed value of the selected entry of the second data set is in the hash table, the hashed value of the selected entry of the second data set;
>
> (i) program instructions that continue (a) through (d) and (e) through (h) respectively for all entries in the first and the second data sets until both the first and the second data sets have been completely processed; and
>
> (j) program instructions that report a difference between the first data set and the second data set in response to at least one hashed value remaining in the hash table.

Additionally, as noted above with regard to claims 1 and 22, neither Orwant nor Musser nor Lou nor Rsync teach or suggest Applicant's claimed novel and non-obvious **removing *the hashed value* of the selected entry of the first data set from the hash table in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table**. As such, because claims 1 and 22 comprise similar limitations of claim 37 not shown or suggested by any prior art reference (or combination thereof), Applicant respectfully urges that Rsync, taken singly or in any combination with Orwant and/or Musser and/or Lou, fails to render the presently claimed invention obvious under 35 U.S.C. §103(a). Specifically, Rsync and/or Orwant and/or Musser and/or Lou, taken singly or in any combination, do not disclose, teach or suggest Applicant's claimed novel and non-obvious use of **program instructions that remove from the hash table, in response to determining that *the hashed value* of the selected entry of the first data set is in the hash table, *the hashed value* of the selected entry of the first data set**.

Applicant's Interpretation of the Prior Art

Applicant's interpretation of the prior art was derived, in part, from the following excerpts:

Orwant

Set difference is noncommutative or asymmetric: that is, if you exchange the order of the sets, the result will change. (Section 6.4.1)

Musser

There is another case in which hash tables are less set-like than sorted associative tables. (page 8, paragraph 9)

Rsync

This tells rsync to copy directories recursively. (OPTIONS section)

Lou

In operation 804, a first tag, that represents a first test case from the second test report, is selected. The selected tag includes several corresponding tags such as a description tag and a results tag. The **description tag** includes a value that describes the test. By way of example, the value of the description tag can be "mousebuttontest". The **results tag** includes a **results value** that identifies the results of the test (e.g., pass, fail, 1, 0, or other test result output value). The value of the description tag that corresponds to the selected tag is used as a key value, in one embodiment. The value of the description tag that corresponds to the selected tag is hashed in operation 806 to create a first hashcode. In one embodiment, the same hash function is used to hash both the first and second test reports. (col. 9, lines 42-55) (emphasis added)

In operation 810, the value stored in the identified location in the hash table is compared to the corresponding results value of the selected tag. If the results value corresponding to the selected tag is not equal to the value in the identified location, the process continues in operation 812 and therefore the results of the common test case are different in the second report than in the first test report. (col. 10, lines 11-20) (emphasis added)

Returning to operation 810, if the value of the selected tag is equal to the value in the identified location in the hash table, the common test case has the same value in both the first test report and the second test report. If the common test case has the same value in both the first test report and the second test report, the process continues in operation 814. In op-

> eration 814, the data (i.e., **the corresponding results value**) is deleted from the identified location in the hash table and the process continues in operation 818. (col. 10, lines 21-30) (emphasis added)

Conclusion

All new claims and/or claim amendments are believed to be fully supported by Applicant's specification.

All independent claims are believed to be in condition for allowance.

All dependent claims are believed to be dependent from allowable independent claims, and therefore in condition for allowance.

Favorable action is respectfully solicited.

Please charge any additional fee occasioned by this paper to our Deposit Account No. 03-1237.

Respectfully submitted,

/Michael T. Abramson/
Michael T. Abramson
Reg. No. 60,320
CESARI AND MCKENNA, LLP
88 BLACK FALCON AVENUE
BOSTON, MA 02210
Telephone: (617) 951-2500
Facsimile: (617) 951-3927